5.4 Data link layer. The Data Link layer in TACO2 is divided into three sublayers: Header Abbreviation, FEC, and Framing. (Effectivity 6: a Medium Access Control Layer, just below the Framing Sublayer, is under consideration.)

5.4.1 Header abbreviation sublayer. TACO2 provides a mechanism for header abbreviation across point-to-point links. Using the header abbreviation sublayer is optional; its inclusion in any compliant implementation of TACO2 shall be mandatory (Effectivity 3). The normal size of a combined NETBLT/IP header for a DATA or LDATA packet is 52 bytes. The overhead due to this header size is significant when small packets are used. TACO2 therefore provides an option for DATA/LDATA packet header abbreviation that reduces the size of headers to eight bytes. The abbreviation mechanism takes advantage of the fact that some header fields are generous in size, and others rarely or never change. A header abbreviated by the sender, in conjunction with state information stored by the receiver, provides all the information necessary to reconstruct the original header at the receiver. 5.4.1.1 Abbreviated header format. The format of a TACO2 packet with abbreviated header is illustrated on figure 18.

FIGURE 18. Abbreviated header TACO2 packet.

The first two bits of the packet are 10. The remaining fields are defined as follows:

a. LP: if this bit is 1, this is the last packet in a buffer (such as, a abbreviated LDATA packet).

b. LB: if this bit is 1, the buffer that this packet belongs to is the last buffer in a transfer.

c. HiConSeqNo: the low-order four bits of the High Consecutive Sequence Number field from the DATA/LDATA packet.

d. Packet Number: the low-order eight bits of the Packet Number field from the DATA/LDATA packet.

e. Buffer Number: the low-order eight bits of the Buffer Number field from the DATA/LDATA packet.

f. Last Buffer Touched: the low-order eight bits of the Last Buffer Touched field from the DATA/LDATA packet.

g. Connection No: the number of the connection to which this packet belongs. In TACO2 implementations that support only one connection at a time, this field shall be 0.

h. Packet Length: the total length (header plus data) of the abbreviated packet in bytes.

i. Checksum: to generate the checksum, the checksum field itself is cleared, the 16-bit ones-complement sum is computed over the abbreviated packet, and the ones complement of this sum is placed in the checksum field (see 5.2.4 for a discussion of checksumming).

j. Data: the Data portion of the original DATA/LDATA packet.

5.4.1.2 Multiple-connection operation with abbreviated headers. TACO2 provides two compatible mechanisms for operation with abbreviated headers. The first, described in this section, allows multiple connections to operate concurrently across a single point-to-point link. The second, described in 5.4.1.3, allows only a single connection at a time across a point-to-point link.

5.4.1.2.1 Sender operation with abbreviated headers. A sender shall provide the option of operation with or without abbreviated headers. If operation with abbreviated headers is selected, the sender shall abbreviate DATA/LDATA packets for an open connection.

5.4.1.2.1.1 Sender connection state table. A sender shall maintain the following information about each connection. The following description, in terms of indexing state table entries, is not intended to dictate an implementation. It describes operation in terms of a connection state table, with space for 16 entries. Each entry includes fields for the following:

a. Source IP address.

b. Destination IP address.

c. Local Port number.

d. Foreign Port number.

e. Connection UID.

f. Burst Size.

g. Burst Interval.

The index of the entry with matching field values shall be used as the abbreviated packet's Connection Number, as explained below.

5.4.1.2.1.2 Sender processing of outgoing OPEN packet. When a sender with header abbreviation turned on detects a NETBLT OPEN packet, it shall examine its connection state table for an entry with the same IP addresses, Local Port number, and Connection UID. If no such entry is found, it shall establish a table entry with those values, using the least recently used entry if no unused entries are available. (Note that reuse of the state table entry for an open connection will make it impossible to abbreviate any more packets for that connection.) The index of the entry with the same IP addresses, Local Port number, and Connection UID shall be used as the connection number. The connection number shall be inserted into the high-order four bits of the Reserved field (referred to in the following sections as the Connection Number field for OPEN and RESPONSE packets) in the OPEN packet, and the packet sent on to the next lower layer. The modified OPEN packet is shown below.

5.4.1.2.1.3 Sender processing of incoming RESPONSE packet. When a sender with header abbreviation turned on detects a NETBLT RESPONSE packet, it shall examine the entry in its connection state table identified by the value of the Connection Number field for IP source and destination addresses equal to the packet's destination and source addresses, Local Port number equal to the packet's Foreign Port number, and the same Connection UID. If the values match, the Local Port number, burst size, and burst interval in the RESPONSE packet shall replace the Foreign Port number, burst size, and burst interval in the table entry. The packet shall be sent on to the IP layer.

5.4.1.2.1.4 Sender processing of outgoing DATA/LDATA packet. When a sender with header abbreviation turned on detects a NETBLT DATA/LDATA packet, it shall examine the connection state table for an entry with the same IP addresses, port numbers, Burst Size, and Burst Interval. If such an entry is found, it shall abbreviate the packet using the format defined in 5.4.1.1, using the index of the entry as the Connection Number, and send it on to the next lower layer. If no such entry is found, the packet shall be sent on unchanged. If a connection state table entry is found with the same IP addresses and port numbers, the burst size and burst interval values in the packet shall replace those in the table entry.

5.4.1.2.2 Receiver operation with abbreviated headers. The receiver shall check the high-order two bits of each received packet to determine the packet type. If the bits are 01, the packet shall be passed unchanged to the IP layer. If the bits are 10, the packet shall be reconstructed by the Header Abbreviation sublayer; the Header Abbreviation sublayer shall pass the reconstructed packet to the IP layer.

5.4.1.2.2.1 Receiver connection state table. The receiver shall maintain a connection state table, with space for 16 entries, containing information needed for correct reconstruction of the reconstructed packet. Each entry shall include fields for the following:

a. Source IP address.

b. Destination IP address.

c. Local Port number.

d. Foreign Port number.

e. Connection UID.

f. Burst Size.

g. Burst Interval.

h. Buffer Number.

i. Last Buffer Touched.

j. High Consecutive Sequence Number Received.

The abbreviated packet's Connection Number shall be used as an index into the connection state table, as explained below.

5.4.1.2.2.2 Receiver processing of incoming OPEN packet. When a receiver detects a NETBLT OPEN packet, it shall fill in the IP address, Local Port number, and Connection UID fields in the connection state table entry indexed by the packet's Connection Number with the values in the packet.

5.4.1.2.2.3 Receiver processing of outgoing RESPONSE packet. When a receiver detects a NETBLT RESPONSE packet, it shall examine its connection state table for an entry with IP source and destination addresses equal to the packet's destination and source addresses, Local Port number equal to the packet's Foreign Port number, and the same Connection UID. If the values match, the Local Port number, burst size, and burst interval in the RESPONSE packet shall replace the Foreign Port number, burst size, and burst interval in the table entry, and the Connection Number field in the RESPONSE packet shall be filled in with the index of the matching connection state table entry.

5.4.1.2.2.4 Receiver processing of incoming abbreviated header packet. When a receiver detects a packet with abbreviated header, it shall use the values in the connection state table entry indexed by the Connection Number to determine the IP addresses, port numbers, burst size, and burst interval in the reconstructed packet. The remaining IP header fields shall be filled in accordance with 5.3.3. The NETBLT Buffer Number, Last Buffer Touched, and High Consecutive Sequence Number Received fields shall be filled in with values computed by combining the abbreviated header value of the corresponding field and the receiver's state table value. For the Last Buffer Touched and High Consecutive Sequence Number Received fields, the computed value shall contain the abbreviated header value for the low-order portion, and the smallest value such that the result is no less than the old state table value for the high-order portion. For the Buffer Number field, the computed value shall use the abbreviated header value for the low-order portion, and a value that causes minimal change between the old state table value and the new computed value for the high-order portion. The state table value shall be changed to correspond to the new computed value. The NETBLT Packet Number field shall be filled in with the abbreviated header Packet Number field, padded with zeros on the left. The NETBLT "L" bit shall be set according to the "LB" bit in the abbreviated header, and the NETBLT Type field shall be DATA or LDATA according to the "LP" bit in the abbreviated header. The remaining NETBLT fields shall be filled in accordance with 5.2.

5.4.1.2.2.5 Receiver processing of incoming DATA/LDATA packet. When a receiver detects a NETBLT DATA/LDATA packet, it shall examine the connection state table for an entry with the same IP addresses and port numbers. If such an entry is found, it shall replace the burst size and burst interval fields in that entry with the values from the DATA/LDATA packet. The packet shall be sent to the IP layer unchanged.

5.4.1.3 Single-connection operation with abbreviated headers. In single-connection operation of TACO2 with abbreviated headers, the connection number shall be ignored. Operation shall be as described in 5.4.1.2, except that the connection state table shall contain only one entry, with the index zero. However, since TACO2 does not mandate interlayer interfaces, it may be implemented in a simpler way, as described in the following.

5.4.1.3.1 Single-connection sender operation with abbreviated headers. The sender shall provide the option of operation with or without abbreviated headers. If operation with abbreviated headers is selected, the sender shall abbreviate DATA/LDATA. The sender shall maintain a set of connection state variables for the following:

a. Burst Size.

b. Burst Interval.

5.4.1.3.1.1 Sender processing of outgoing OPEN packet. In single-connection operation, the sender with header abbreviation turned on shall store the Burst Size and Burst Interval in the connection state variables and pass the OPEN packet unchanged to the next lower layer.

5.4.1.3.1.2 Sender processing of incoming RESPONSE packet. In single-connection operation, the sender with header abbreviation turned on shall store the Burst Size and Burst Interval in the connection state variables and pass RESPONSE packet unchanged to the next higher layer.

5.4.1.3.1.3 Sender processing of outgoing DATA and LDATA packets. In single-connection operation, the sender with header abbreviation turned on shall compare the DATA/LDATA packet's Burst Size and Burst Interval with the values stored in the connection state variables. If both match, it shall abbreviate the packet using the format defined in 5.4.1.1, using zero as the Connection Number. If either variable does not match, the DATA/LDATA packet shall be sent on unchanged, and the burst size and burst interval values in the packet shall replace those in the table entry.

5.4.1.3.2 Single-connection receiver operation with abbreviated headers. The receiver shall check the high-order two bits of each received packet to determine the packet type. If the bits are 01, the packet shall be passed unchanged to the IP layer. If the bits are 10, the packet shall be reconstructed by the Header Abbreviation sublayer ; the Header Abbreviation sublayer shall pass the reconstructed packet to the IP layer. The receiver shall maintain a set of connection state variables, corresponding in size to those in a DATA/LDATA packet, for the following:

a. Buffer Number.

b. Last Buffer Touched.

c. High Consecutive Sequence Number Received.

d. Burst Size.

e. Burst Interval.

f. Local Port.

g. Foreign Port.

5.4.1.3.2.1 Receiver processing of incoming OPEN packet. In single-connection operation, upon detection of an OPEN packet, the receiver shall set the first three connection state variables to zero, store the Burst Size, Burst Interval, Local Port, and Foreign Port, and pass the OPEN packet unchanged to the IP layer.

5.4.1.3.2.2 Receiver processing of outgoing RESPONSE packets. In single-connection operation, the receiver shall store the Burst Size, Burst Interval, Local Port in the Foreign Port variable, and Foreign Port in the Local Port variable, and pass the packet unchanged to the next lower layer.

5.4.1.3.2.3 Receiver processing of incoming abbreviated-header packets. In single-connection operation, the receiver shall reconstruct a DATA/LDATA packet from the incoming abbreviated-header packet . The NETBLT Buffer Number, Last Buffer Touched, and High Consecutive Sequence Number Received fields shall be filled in with values computed by combining the abbreviated header value of the corresponding field and the receiver's connection state variable value. For the Last Buffer Touched and High Consecutive Sequence Number Received fields, the computed value shall contain the abbreviated header value for the low-order portion, and the smallest value such that the result is no less than the old connection state variable value for the high-order portion. For the Buffer Number field, the computed value shall use the abbreviated header value for the low-order portion, and a value that causes minimal change between the old connection state variable value and the new computed value for the high-order portion. The connection state variable value shall be changed to correspond to the new computed value. The NETBLT Packet Number field shall be filled in with the abbreviated header Packet Number field, padded with zeros on the left. The NETBLT "L" bit shall be set according to the "LB" bit in the abbreviated header, and the NETBLT Type field shall be DATA or LDATA according to the "LP" bit in the abbreviated header. Burst Size and Burst Interval shall be filled in according to the variable values. The remaining NETBLT fields shall be filled in accordance with 5.2. The reconstructed NETBLT packet shall be passed directly to the NETBLT layer (bypassing the IP layer) for normal processing.

5.4.2 FEC sublayer. Using the FEC sublayer is optional; the inclusion of FEC-I in any compliant implementation of TACO2 is mandatory.

5.4.2.1 FEC-I code. The FEC-I encoding process takes each IP datagram to be transmitted, adds Reed-Solomon redundancy, and passes the encoded datagram to the link layer for encapsulation, generally as an HDLC frame as specified in 5.4.3.1. As a result, the HDLC implementation shall (TBR) allow received packets to be processed by the FEC sublayer even if the HDLC checksum is in error. The encoded datagram also may be encapsulated as a SLIP frame, as specified in 5.4.3.2. If the unencoded datagram contains K bytes where K is not greater than 152, the encoded datagram contains a single Reed-Solomon codeword containing K + 10 bytes. For purposes of Reed-Solomon code definition, these bytes are numbered from 0 to K + 9 as shown on figure 19 (where the left end of the figure represents the beginning of the datagram if viewed in time-sequence).

FIGURE 19. FEC-I format.

The FEC-I code uses arithmetic in the Galois field GF(256), as specified in the standard ISO 9171 for 5.25" WORM disk coding. This field has 256 elements, which are represented by 8-bit symbols ("octets" or simply "bytes"), using the generator polynomial $x^8 + x^5 + x^3 + x^2 + 1$. The primitive element a has hexadecimal value 0x02. The Reed-Solomon check bytes Cj are defined by the following:

5.4.2.1.1 Correction capability. Each codeword as defined in 5.4.2.1 has distance 11 and is fully correctable with up to five independent byte-errors. FEC-I can fully correct all patterns of five or fewer erroneous bytes in any codeword. Note that the content and sequence of the message bytes M_i remains unchanged by the encoding process.

5.4.2.1.2 Long datagrams. If the length of the datagram to be encoded is greater than 152 bytes but does not exceed 752 bytes, encoding is performed by including up to five separate concatenated Reed-Solomon codewords in the encoded datagram. The maximum encoded length is 802 bytes. As an example, figure 20 shows how a datagram with an unencoded length of 450 bytes would be encoded, giving a datagram 480 bytes long.

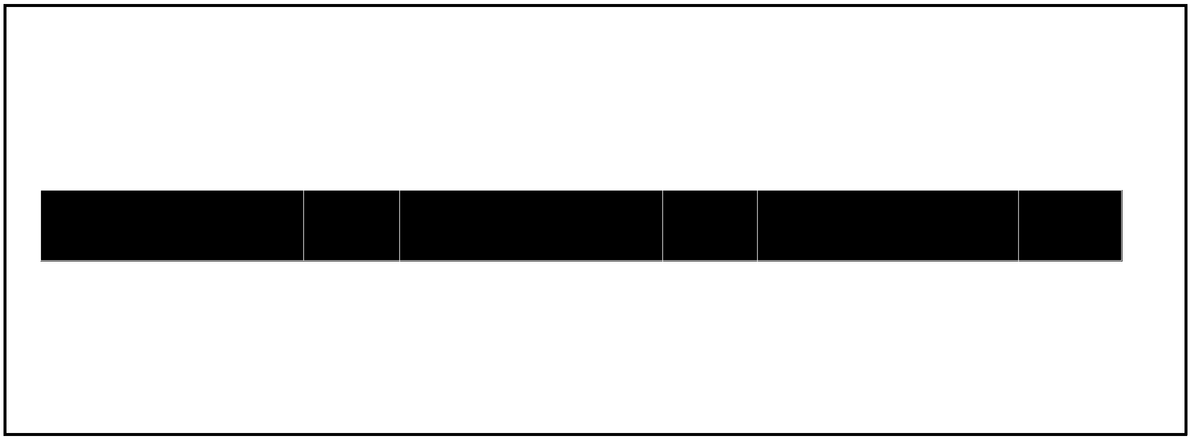

FIGURE 20. Encoding a 450 byte packet.

As shown on figure 20, only the final codeword in a multicodeword encoded datagram may be truncated to fewer than 152 message bytes. The time sequence of the message bytes is unchanged by the encoding process. FEC-I encoding is presently not specified for datagrams whose unencoded length is greater than 752 bytes. Should a FEC-I encoder be presented with such a datagram, the correct action is to transmit it without any encoding.

5.4.2.2 Required modes of FEC. The following mandatory FEC modes (see 5.4.2.2.1 and 5.4.2.2.2) shall be supplied by Secondary Imagery Dissemination (SID) devices. Each of these three modes shall be able to bypass the FEC capability contained in any datalink hardware external to the SID device. The operational selection of the three modes (see 5.4.2.2.1, 5.4.2.2.2, and 5.4.2.2.3) shall be controllable using the SIDS user interface.

5.4.2.2.1 Uncoded. The SLIP and/or HDLC encapsulated datalink is provided with no FEC coding applied by the SIDS system.

5.4.2.2.2 FEC-I. FEC-1 is applied to a SLIP and/or HDLC encapsulated datalink as described in 5.4.2.1.

5.4.2.2.3 FEC-II. FEC-II is applied to a SLIP and/or HDLC encapsulated datalink as described in appendix C (Effectivity 2).

5.4.2.3 Bit error ratio testing (BERT).

5.4.2.3.1 Bit error ratio test facility. The Data Link Layer shall provide the upper protocol layers with a Bit Error Ratio Test facility. The services provided to the upper layer are:

a. The ability to send a group of BERT frames over the link.

b. The ability to notify that a group of BERT frames has been received, along with the following information: number of frames successfully received; the IP address of the station originating the BERT test; and whether the received frames form a standard BERT test or a short BERT test as defined below.

5.4.2.3.2 BERT frame format. The BERT frame may be encapsulated by either the HDLC protocol or the SLIP protocol. If encapsulated by HDLC, the BERT frame shall contain 12 data bytes followed by the 2-byte frame check sequence as defined by the HDLC protocol. If encapsulated by SLIP, the BERT frame shall contain only the 12 data bytes. In either case, the 12 data bytes, numbered 0 through 11 in time sequence, shall be formatted as shown on figure 21.

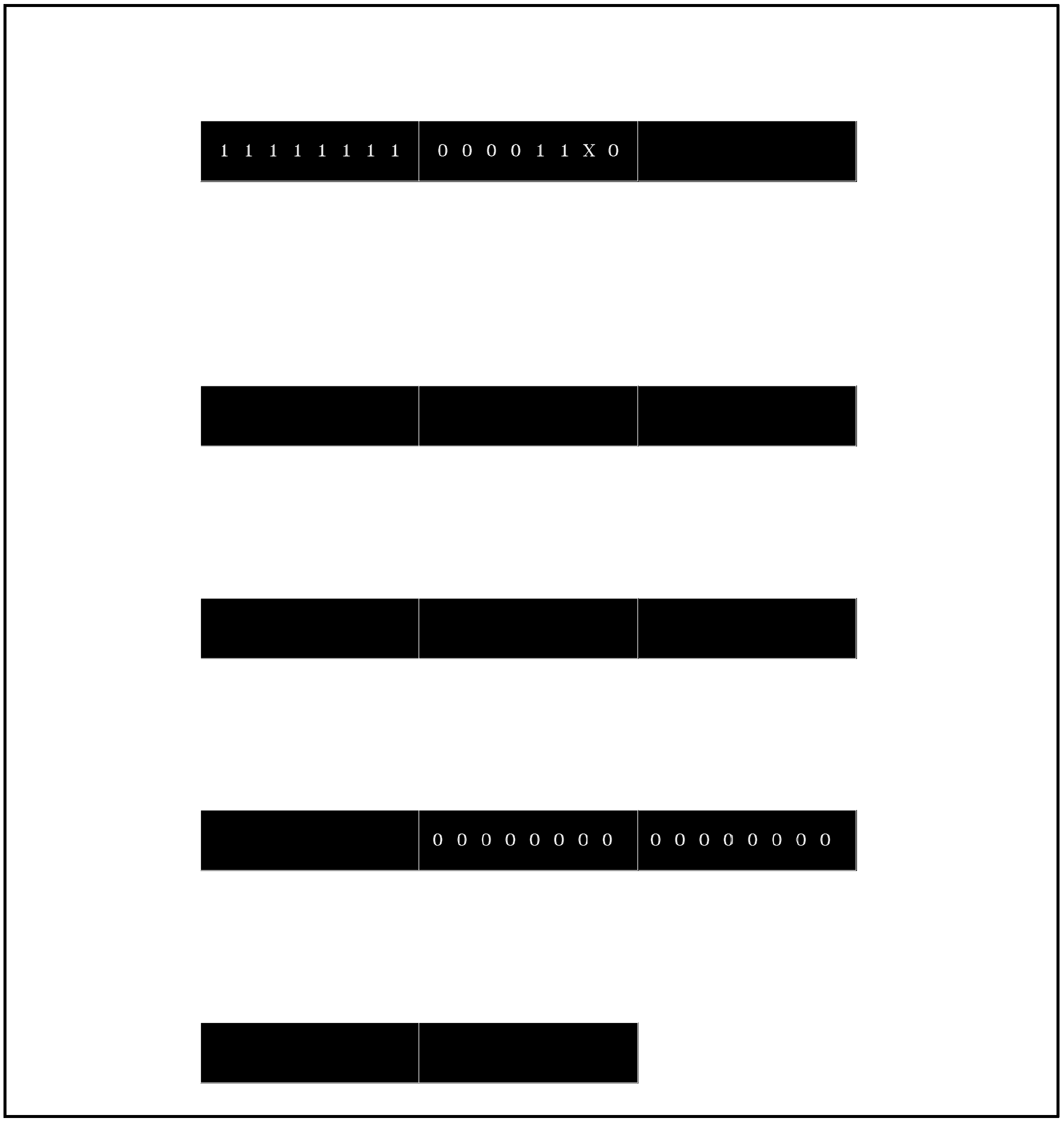

FIGURE 21. BERT frame format.

5.4.2.3.3 Standard BERT test format. The standard BERT test shall consist of sending 1,000 (one thousand) identical BERT frames, each with the hexadecimal character "0x0E" in byte number 1.

5.4.2.3.4 Short BERT test format. The short BERT test shall consist of sending 200 (two hundred) identical BERT frames, each with the hexadecimal character "0x0C" in byte number 1.

5.4.3 Framing sublayer. Two Framing sublayers are included in the TACO2 protocol stack: HDLC Framing for synchronous operation, and SLIP for asynchronous operation. HDLC Framing capability shall be available in any compliant TACO2 implementation; the implementation of SLIP is optional.

5.4.3.1 HDLC framing.

5.4.3.1.1 Overview. The synchronous data link layer for TACO2 uses HDLC framing as a standard transparent encapsulation for the IP packets supplied by the next higher protocol layer. TACO2 uses only the frame structure, and not the control procedures, of HDLC, the ISO High-level Data Link Control. The frame structure used is compatible with ISO 3309-1979 and with the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation X.25, which is based on HDLC.

5.4.3.1.2 Required HDLC components. The standard HDLC frame structure is shown below. The fields shall be transmitted from left to right.

5.4.3.1.2.1 Flag sequence. The flag sequence shall be a single octet that indicates the beginning or end of a frame. The flag sequence consists of the binary sequence 01111110 (hexadecimal 0x7E). A flag that terminates one frame may also signal the beginning of the next frame, which allows back-to-back frames to be separated by a single flag.

5.4.3.1.2.2 Address field. The address field shall be a single octet. It shall contain 0xFF if no FEC coding is applied by the FEC sublayer, and shall contain 0x33 if FEC-1 coding, as specified in 5.4.2.1, is applied.

5.4.3.1.2.3 Control field. The control field shall be a single octet that indicates the type of frame. The control field for TACO2 shall contain the binary sequence 00000011 (hexadecimal 0x03), the Unnumbered Information (UI) command with the Poll/Final (P/F) bit set to zero.

5.4.3.1.2.4 Information field. The information field contains data for the next higher protocol layer, which for TACO2 shall be an IP packet. The end of the information field is found by locating the closing flag and allowing two octets for the Frame Check Sequence field. The Information field shall be an integer number of octets in length.

5.4.3.1.2.5 Frame check sequence field. The Frame Check Sequence (FCS) field is two octets. It shall be calculated over all bits of the address, control, and information fields, not including any bits inserted for transparency. This does not include the flag sequences or the FCS field. The polynomial used is $x^{16}+ x^{12}+x^5+1$. This is the standard CCITT Cyclic Redundancy Check (CRC) polynomial.

5.4.3.1.3 HDLC procedures.

5.4.3.1.3.1 Order of bit transmission. The information field shall be transmitted with the high-order (most significant) octet first. Individual octets of all fields except the FCS, but including the Information field, shall be transmitted low-order bit first. The FCS shall be transmitted with the coefficient of the highest term first.

5.4.3.1.3.2 Transparency. Bit-stuffing is used to distinguish the flag sequence from other fields. The transmitter shall insert a 0 bit after all non-flag sequences of 5 contiguous 1 bits. The receiver shall discard a 0 bit received after five contiguous 1 bits. The reception of a sixth 1 bit indicates a flag when followed by a zero bit, or an abort when followed by a seventh 1 bit.

5.4.3.1.3.3 Invalid frames. In duplex mode with the optional FEC sublayer not operating, invalid frames shall be discarded. An invalid frame is one that is too short, too long, contains a nonintegral number of octets, contains an unrecognized address or control field, has an invalid FCS, or was aborted. In simplex mode and in duplex mode with the FEC sublayer operating, a frame with an invalid FCS that is otherwise valid may be passed on to the FEC sublayer. A frame of less than four octets (excluding flags) is too short; a TACO2 implementation shall support a maximum frame size of at least 576 information octets.

5.4.3.1.3.4 Frame abortion. A frame may be aborted by transmitting at least 7 contiguous 1 bits.

5.4.3.1.3.5 Inter-frame time fill. Inter-frame time fill, when required within a period of continuous transmission, shall be accomplished by transmitting flag sequences. The transmitted level for the interval between periods of continuous transmission shall be in accordance with the requirements of the attached cryptographic device, if any.

5.4.3.2 SLIP. Devices using asynchronous communications shall use SLIP.

5.4.3.2.1 Overview. The asynchronous data link layer for TACO2 uses SLIP as a standard transparent encapsulation for the IP packets supplied by the next higher protocol layer. SLIP defines a sequence of characters that frame packets on a serial line in a simple and consistent manner.

5.4.3.2.2 Protocol. The SLIP protocol defines two special characters: END and ESC. END is hexadecimal 0xC0 (decimal 192) and ESC is hexadecimal 0xDB (decimal 219), not to be confused with the ASCII ESC character. For this discussion, ESC will indicate the SLIP ESC character. To send a packet, a SLIP host shall send an END character followed by the data in the packet. If a data byte is the same code as the END character, a two-byte sequence of ESC and hexadecimal 0xDC (decimal 220) shall be sent instead. If a data byte is the same code as the ESC character, a two byte sequence of ESC and hexadecimal 0xDD (decimal 221) shall be sent instead. When the last byte in the packet has been sent, an END character shall be transmitted.

5.4.3.2.3 Required SLIP components. The only required SLIP components are the mechanisms for indicating End of Packet and for byte-stuffing.

5.4.3.2.4 Specific values for SLIP.

5.4.3.2.4.1 Order of bit transmission. Information shall be transmitted with the high-order (most significant) octet first. Individual octets shall be transferred low-order bit first. This is the standard asynchronous transmission order.

5.4.3.2.4.2 Transparency. Byte-stuffing shall be used to distinguish the End of Packet character from the same value when it occurs in the information being transmitted.

5.4.3.2.4.3 Invalid frames. The only invalid frames in SLIP shall be ones that are too long. The receiver shall be prepared to accept frames of at least 576 octets (not including framing and byte-stuffing characters), and preferably of at least 2048 octets. Frames that are too long may be discarded or truncated.

5.4.3.2.4.4 Inter-frame gap. Frames may be separated by a single END character. Any inter-frame gap shall consist of continuous marks or continuous END characters.

5.5 DTE-DCE interfaces. TACO2 does not specify a single standard Data Terminal Equipment - Data Circuit-terminating Equipment (DTE-DCE) interface. Detailed implementation information for certain communications environments (such as, recommended parameter values, interfaces, and settings/strappings for KY-57, KG-84A/C, KY-68 and STU-III cryptographic devices) is provided by the JIEO TIS listed in section 2. Additional TIS documents will be prepared for other communications environments. TACO2 implementations shall provide DTE-DCE interfaces capable of conforming to the TIS recommendations for each cryptographic device or communications environment identified by the procuring activity.